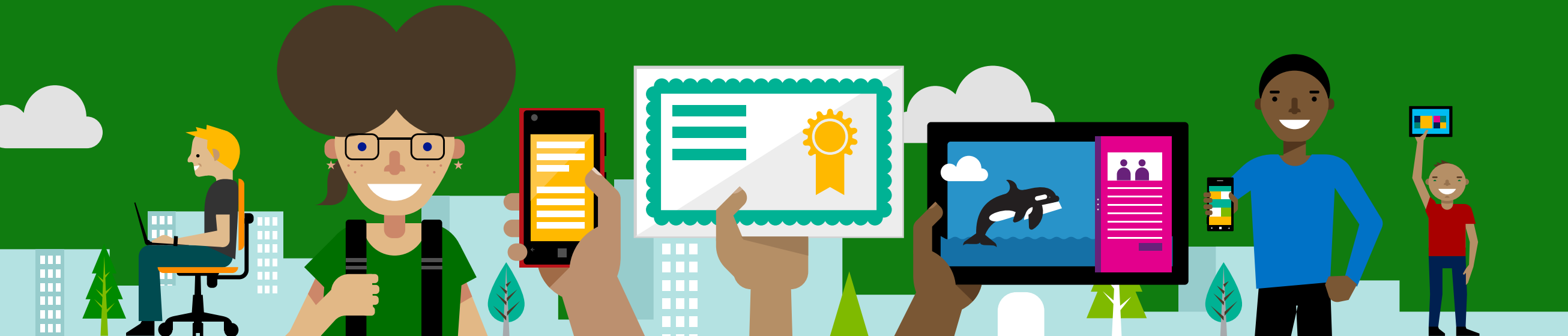


Managing the Graphical Interface by Using CSS

3.3. Managing the Graphical Interface by Using CSS.



Agenda

1

Graphic Effects

2

2D and 3D Transformations



Graphic Effects



New Effects in CSS3



I have rounded corners!!!



I'm 50% transparent!



I have a shadow!



I'm animated!

- New additions in CSS3 allow you to create a more engaging experience for users
- 2D and 3D animations and transformations make moving from page to page more fun
- Drop shadows and rounded corners make text and graphics more visually appealing

Creating Rounded Corners

In CSS3, you can now add rounded corners to layout elements, including headers, footers, sidebars, and images

Use the `border-radius` property along with a length value, such as pixels, ems, or a percentage

- the higher the value, the more rounded a corner will be
- some browsers have problems rendering a percentage value

You can round individual corners with the following properties:

- `border-top-left-radius`
- `border-top-right-radius`
- `border-bottom-right-radius`
- `border-bottom-left-radius`

```
.rounded {  
    border-radius: 20px 20px 20px  
    20px;  
}
```

I have rounded corners!!!

Creating Shadows

- Add drop shadows to elements using the `box-shadow` property
 - a drop shadow is a visual effect that makes an object appear like it is floating above other objects on a document
- There are six attributes that can be modified when using the `box-shadow` property: `h-shadow`, `v-shadow`, `blur`, `spread`, `color`, and `inset`
- The `h-shadow` and `v-shadow` attributes must be included, while the others are optional

`box-shadow: 10px 10px 8px #808080;`

h-shadow v-shadow blur color

box-shadow Attributes

ATTRIBUTES	DESCRIPTION
h-shadow	Defines the horizontal offset of the shadow. A positive value places the shadow to the right of the object, while a negative value places it to the left.
v-shadow	Defines the vertical offset of the shadow. A positive value places the shadow above the object, while a negative value places it below.
blur	Defines the amount of blur applied to the shadow.
spread	Determines the size of the shadow.
color	Determines the color of the shadow.
inset	Moves the shadow to the inside of the box.

Applying Transparency

Apply the effect of transparency with the `opacity` property

- Opacity determines how transparent an object appears

The value for the `opacity` object is a number between 0.0 and 1.0

- 1.0 means that the object is entirely opaque, while 0.0 means that it is entirely transparent

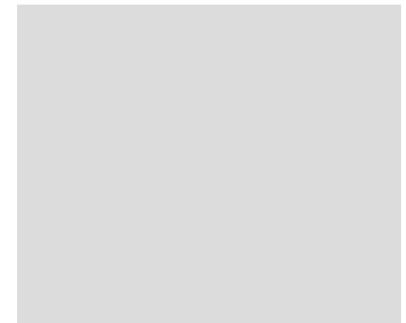
1.0



0.5



0.2



Rounded Corners, Shadows, and Transparency Demo

```
.rounded {  
    border-radius: 20px 20px 20px  
20px;  
    box-shadow: 10px 10px 8px  
#808080;  
    opacity: .5;  
}
```

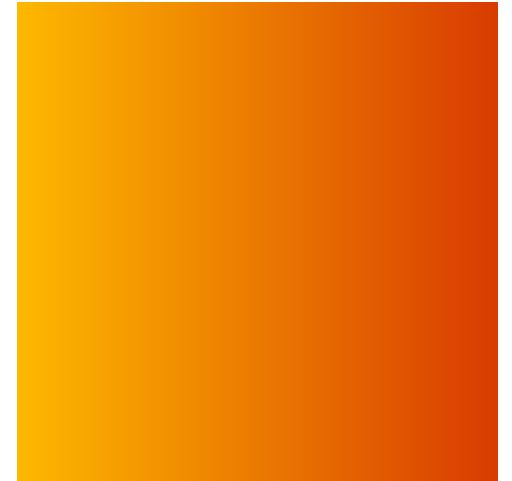
Background Gradients

- A gradient is a smooth transition from one color to another
- In CSS3, you can apply gradients to a number of objects, such as headers, footers, buttons, and more
- Gradients are applied through the background property with one of the methods featured in the table to the right

GRADIENT METHODS	DESCRIPTION
linear-gradient	Creates a gradient from top to bottom or vice versa, or from corner to corner
radial-gradient	Creates a gradient that radiates out from the center
repeating-linear-gradient	Repeats a linear gradient
repeating-radial-gradient	Repeats a radial gradient with alternating bands of gradient color

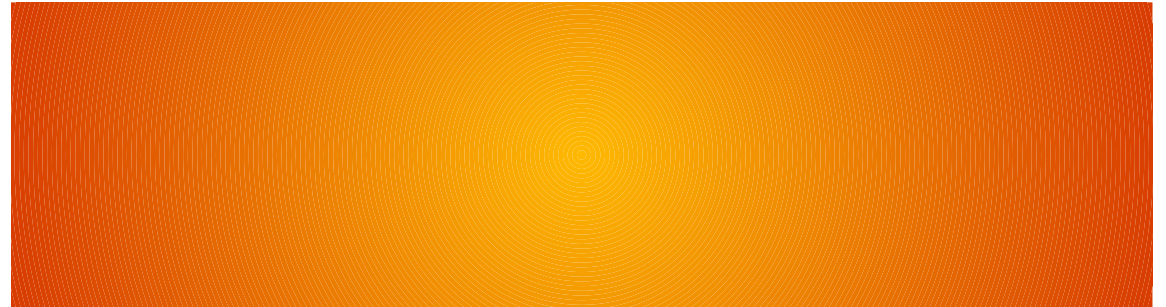
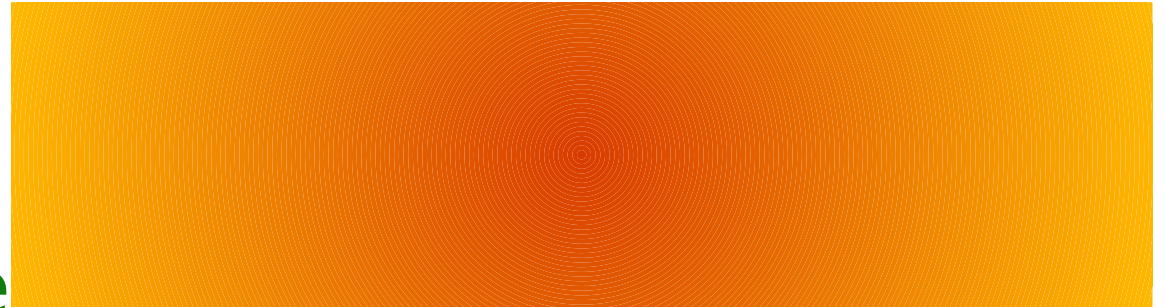
Linear Gradients

- Linear gradients can be from top to bottom, bottom to top, or from corner to corner
- A default gradient goes from top to bottom
- Insert to top, to bottom, to right, or to left as the first value with the linear-gradient method to control the direction



Radial Gradients

- Radial gradients start from the center of an object and radiate outwards
- The first values used with the radial-gradient method determine the horizontal and vertical center values



Gradients Demo

```
.gradiented {  
    background: linear-gradient(to top, white 20%, blue 80%);  
}
```

2D and 3D Transformations



Transformations

- In HTML5/CSS3, you can now transform 2D and 3D objects
- A **transform** is an effect that lets you change the size, shape, and position of an element
 - This includes rotating, scaling, stretching, spinning and moving elements
- To add transformations, use the transform property with one of its methods

Methods for the Transform Property,

VALUE	DESCRIPTION
<code>matrix (n,n,n,n,n,n)</code>	Specifies a 2D transformation using a six-value matrix
<code>matrix3d (n,n,n,n,n,n,n, n,n,n,n,n,n,n, ,n)</code>	Specifies a 2D transformation using a sixteen-value matrix
<code>perspective(n)</code>	Specifies a perspective view for a 3D element that's been transformed
<code>rotate (angle)</code>	Rotates an element in 2D
<code>rotate3d (x,y,z,angle)</code>	Rotates an element in 3D
<code>rotateX (angle)</code>	Rotates an element in 3D along the x-axis

VALUE	DESCRIPTION
<code>rotateY (angle)</code>	Rotates an element in 3D along the y-axis
<code>rotateZ (angle)</code>	Rotates an element in 3D along the z-axis
<code>scale (x,y)</code>	Scales an element in 2D (width and height)
<code>scale3d (x,y,z)</code>	Scales an element in 3D (width, height, and an arbitrary vector in 3D space)
<code>scaleX (x)</code>	Scales an element in 3D along the x-axis
<code>scaleY (y)</code>	Scales an element in 3D along the y-axis

Methods for the Transform Property,

VALUE	DESCRIPTION
scaleZ (z)	Scales an element in 3D along the z-axis (a vector in 3D space)
skew (x-angle, y-angle)	Skews an element in 2D along the x-axis and the y-axis
skewX (angle)	Skews an element in 3D along the x-axis
skewY (angle)	Skews an element in 3D along the y-axis
translate (x,y)	Translates (moves) an element in 2D
translate3d (x,y,z)	Translates (moves) an element in 3D

VALUE	DESCRIPTION
translateX (x)	Translates an element in 3D using the x-axis
translateY (y)	Translates an element in 3D using the y-axis
translateZ (z)	Translates an element in 3D using the z-axis

2D Translations

- A translation moves an element without rotating, skewing, or turning the image.
- To translate an object, use the transform property in combination with the `translate()` method
- The `translate()` method accepts two values: one to adjust its position on the x-axis and another on the y-axis

```
.translated {  
  transform: translate(200px, 0px);  
}
```

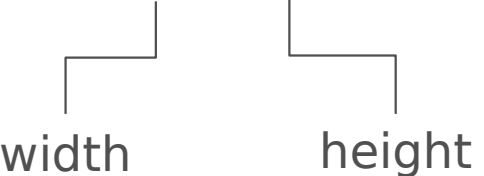


The diagram shows two lines originating from the values '200px' and '0px' in the CSS code. The line from '200px' points down to the label 'x-axis'. The line from '0px' points down to the label 'y-axis'.

2D Scaling

- Scaling an element will increase or decrease its size.
- To scale an object, use the transform property in combination with the `scale()` method
- The `scale()` method accepts two values: a factor to adjust its width and another to adjust its height

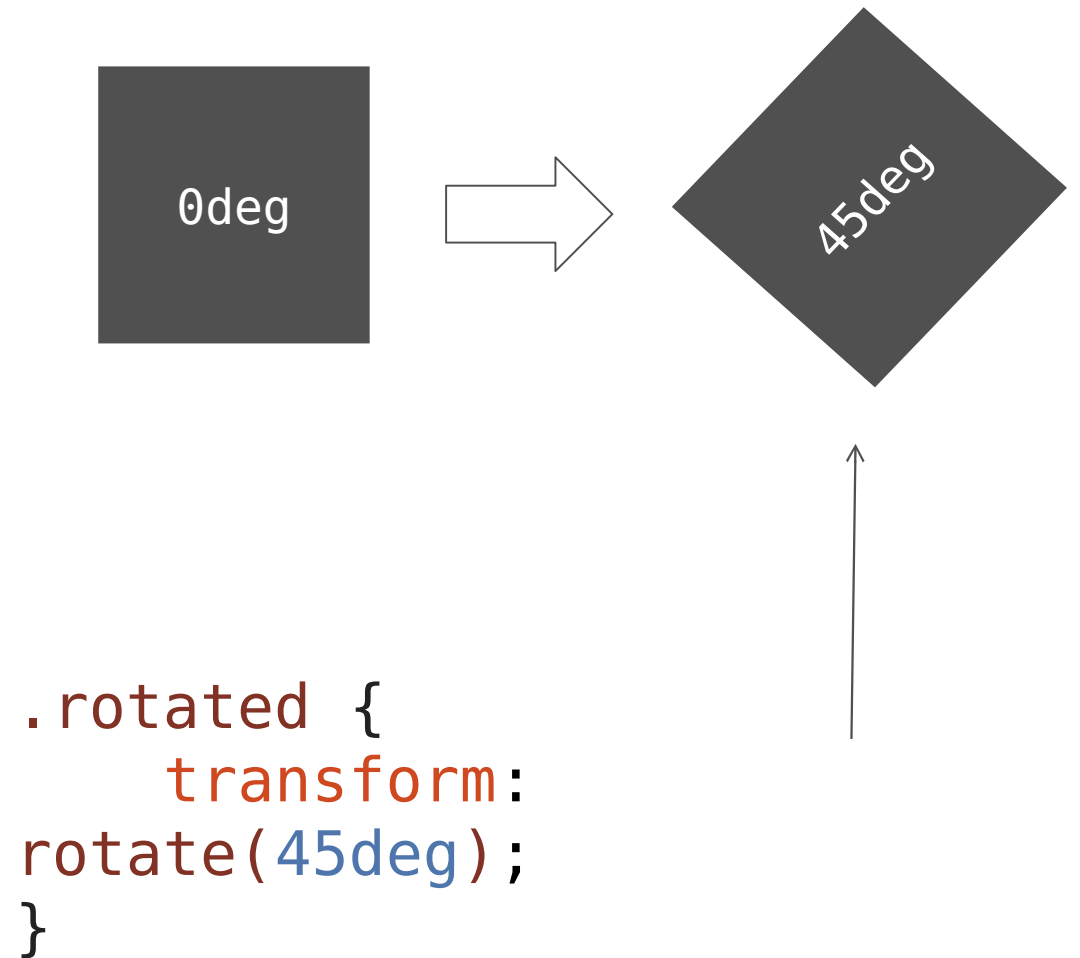
```
.scaled {  
  transform: scale(2, 2);  
}
```



A diagram with two L-shaped brackets pointing to the arguments '2' in the `scale(2, 2);` line of the code block. The bracket under the first '2' is labeled 'width' and the bracket under the second '2' is labeled 'height'.

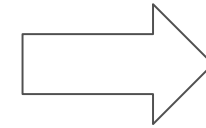
2D Rotation

- Rotating an element will turn it clockwise or counterclockwise
- To rotate an object, use the transform property in combination with the rotate() method
- The rotate() method accepts one value: the number of degrees to rotate the object

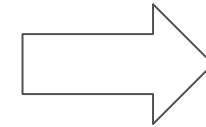


3D Rotation

- 3D rotation will rotate an element around its x-axis or y-axis
- To 3D rotate an object, use the transform property in combination with the rotateX() or rotateY() methods
- The rotateX() and rotateY() methods accept one value: the number of degrees to rotate the object around an axis



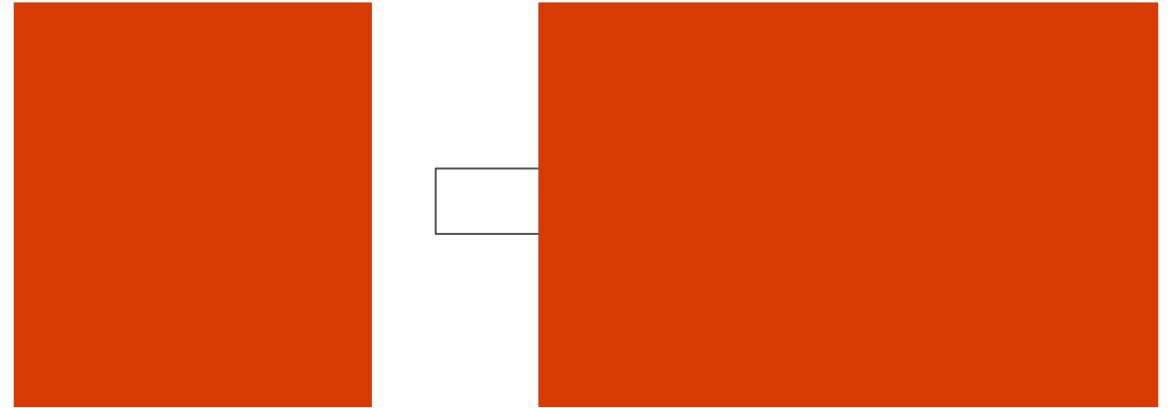
transform: rotateX(30deg);



transform: rotateY(45deg);

2D Skewing

- Skewing an element will stretch it one or more directions
- To skew an object, use the transform property in combination with the skew() method
- The skew() method accepts two values: the number of degrees to turn the object around the x-axis and another for the y-axis



```
.skewed {  
    transform: skew(45deg,  
15deg);  
}
```

Transitions

- In CSS3, a transition is a visible movement from one state to another on screen
- To create a transition, use the `transition` property
 - Set the value as the property that you would like to change
- You must also set the duration, the delay, and the timing function
 - This can be done with the `transition` property or by using specific properties for each (see next slide)

```
.transitioned {  
    transition-  
property:background;  
    transition-duration: 2s; }  
  
.transitioned:hover{  
    background: #ff6a00;}
```

CSS3 Transition Properties

PROPERTY	DESCRIPTION
transition	Is the shorthand way to specify settings for transition- property, transition-delay, transition-duration, and transition-timing-function at once
transition-property	Specifies the CSS properties that are to be transitioned
transition-delay	Specifies the amount of time that passes after the value changes and before the transition starts; in seconds or milliseconds
transition-duration	Specifies the length of the transition in seconds or milliseconds; starts after the transition-delay property
transition-timing-function	Specifies the speed curve of the transition effect; allows a transition to change speed over its duration Default value = ease, which starts relatively fast and slows down toward the end

Animations

- Animations move static images in a way that makes them appear as though they are moving
- Unlike transitions, animations use the `@keyframes` rule, which allows you to adjust the timings of an animation
- You can dictate when an action occurs by using a percent
 - i.e. 50% of the way through this animation, the element should be yellow

```
@keyframes glow {  
  from {  
    background-color: blue;  
  }  
  50% {  
    background-color: yellow;  
  }  
  to {  
    background-color: red;  
  }  
}  
  
.animated {  
  animation-name: glow;  
  animation-duration: 5s;  
}
```

CSS3 Animation Properties, pt. 1

PROPERTY	DEFAULT VALUE	DESCRIPTION
@keyframes		Creates the animation
animation		Shorthand may to specify all animation properties at once, other than the animation-play-state property
animation-name		Specifies the @keyframes animation name
animation-duration	0	Specifies the length of an animation; in seconds or milliseconds
animation-timing-function	1	Specifies how the animation progresses during one cycle
animation-delay	none	Specifies when the animation starts

CSS3 Animation Properties, pt. 2

PROPERTY	DEFAULT VALUE	DESCRIPTION
<code>animation-iteration-count</code>	1	Specifies the number of cycles of an animation
<code>animation-fill-mode</code>	none	Specifies the values applied by the animation outside the time it executes
<code>animation-direction</code>	normal	Specifies whether the animation plays in reverse on alternate cycles
<code>animation-play-state</code>	running	Specifies the state of the animation; values are running or paused
<code>animation-iteration-count</code>	1	Specifies the number of cycles of an animation
<code>animation-fill-mode</code>	none	Specifies the values applied by the animation outside the time it executes

Agenda

1

Graphic Effects

2

2D and 3D Transformations



